MAHARANA MEWAR PUBLIC SCHOOL

Session: 2025-2026

The Messier Dualis

Class:- XII - D

Subject:- Computer Science (083)

Board Roll No. :-

Submitted By:- Taha Husain

Submitted to:- Mrs. Payal Maheshwari (PGT - CS)

Table of Contents

1. Certificate
2. Acknowledgement
3. Introduction
4. Aim
5. Code(s) of the Project
6. Output(s) of the Project
7. Bibliography/Credits & Conclusion
8. ThankYou!

Certificate

This isto certify that **Taha Husain** of Class XII-D of Maharana Mewar Public School has completed his project file under my supervision. I certify that this project is up to my expectations and as per the guidelines issued by the CBSE.

Signature

(Mrs. Payal Maheshwari, PGT - CS)

Acknowledgement

I take this opportunity to express my profound gratitude and deep regard for my guide's encouragement throughout the course of this project. My thanks and appreciation also go to Mrs. Payal Maheshwari for her amazing guidance, monitoring, and wisdom. I also want to thank my peers who helped in developing the project, and people who have willingly helped me out with their respective abilities.

Introduction

So, For the longest time in my life, I have been in Love with Astronomy and it's sister fields.
Once in 2nd Grade, I had found a YouTube video named 'A Space Journey', and i was in awe from it. That's where my love for Astronomy blossomed (Or atleast that's where i think it did).
And soon enough in Highschool i also developed a hobby for Computers and naturally opted for computer science in my 11th and 12th grades.

At First, I wanted to make a 3D Simulation Software for our school, but the Game Engine for that would not work with Python, and it would take a lot of time to develop it.
So, I combined my love for Space with my Hobby of Computing and made this!

Welcome to the Messier Dualis!

Aim

The Aim of the MessierDualisis to expose people to the beauty of our cosmos and bonus if they are able to leave having learnt a little bit more than before :D

README File

The-Messier-Dualis
(This is the Repository of 'The Messier Dualis' Python Project as my Final Project of my Computer Science Class (Highschool))

Some Info about the Project

'Messier Dualis' is a Interactive GUI Based Project about the Messier List of Astronomical Objects (About 110 Objects) in the night sky.

The Application has HD images of all the Messier Objects (As best as I could fit into here) along with Information about them like (Messier Number, Common Name, Distance from Earth (Light Years), Date of Discovery, Name of the Discoverer, Object Type, Constellation of which it is a part of, Apparent Magnitude, A short Description of it (About 80 words long))

Credit also goes to 'Osric Dienda' for providing the images and some of the Data points

The goal of this project is to make others appreciate the beauty of the cosmos, and a bonus if they could leave having learned a bit more than before!

-> (PS: All the Data points and Images used here are available to download for anyone who wishes to make another project with them)

How to Run: Easy! Just download the Repository, and after extraction and all those shenanigans, Run the 'Main-Program-India.exe' to experience my project!

How to access the Data Points: Easy again! You can either use the '.db' file which is a sqlite3 Database for the Project having all the Data in it. **OR** Just use the '.json' file with all the info in it
(All the images used in this are availible in the Assets Folder)

Controls: Use Right Arrow to move to the next object.
Use Left Arrow to move to the previous object.
Use the Escape key to exit the program

There are 2 Availible Modes of Exploration in it:
1. Randomised Exploration Mode
2. Serial-Wise Exploration Mode

Randomised Exploration mode is where the program chooses random objects for you to explore from the catalogue (Using functions from the python random module)
The Serial-Wise Exploration mode is in which it shows objects in the order that Sir Charles Messier Added them in (1 -> 110)

(There is also some Soothing Music and the occasional Sound Effects availible for some Ear-Candy)

Well Anyways, That's about it for now!

I hope you enjoy my Project and have learnt something new from it :D

By-Bieeeee!! 👋

The Messier Dualis - Source Code

Database_Population_Code-Romeo.py

```
----------------------------------------

from ai import call_gpt
import time
import csv
Main = {}

print("Welcome to the Database Population Code!")
time.sleep(1)

def DataSaving(Desc_Dict):
    filename = "Database_Descriptions.csv"
    csvfile = open(filename, mode = 'w', newline = '')
    writer_lambda = csv.writer(csvfile)
    writer_lambda.writerow(Desc_Dict.keys())
    writer_lambda.writerow(Desc_Dict.values())

    print(f"The Dictionary has been Succesfully Saved as {filename}!")
    csvfile.close()

while(True):
    #print("1. Populate the Years of Discovery to the Database")
    #time.sleep(0.5)
    #print("2. Populate the Names of the Discoverers to the Database")
    #time.sleep(0.5)
    #print("2. Populate Fun Facts of the Objects to the Databse")
    #time.sleep(0.5)
    print("1. Populate the Descriptions of the Objects to the
Database")
    time.sleep(0.5)
    print("2. Populate the Fun Facts of the Objects to the Database")
    time.sleep(0.5)
    print("3. Exit the Program")
    time.sleep(1)

    inp = int(input("Please enter what you wanna do?: "))
    if (inp == 1):
        time.sleep(0.6)
```

```
        print("Alright! Starting the Database Population Code for
Messier Objects Descriptions!")
time.sleep(0.5)
for i in range (1, 111):
            response = call_gpt(f"Give me a Nice, Crisp 80 word
Description for Messier {i} please (Maintain a Tone of Enthusiasim and
make the Descriptions Engaging to read. Also put effort into sounding
Human rather than AI. Also put great effort into ensuring that the
Descriptions are factaully correct)")
            Main[f"Messier {i}"] = response
            print(f"Messier {i} has been added to the Dictionary!")
        time.sleep(1)
        print("All the Messier Object's Information has been added to
the Dictionary!")
        time.sleep(1)
        print(Main)
        time.sleep(4)
        DataSaving(Main)

    elif (inp == 2):
        time.sleep(0.6)
        print("Alright! Starting the Database Population Code for
Messier Objects Fun Facts!")
        time.sleep(0.5)
        for i in range (1, 111):
response = call_gpt(f"Give me a Short Fun Fact for Messier
{i} please (Maintain a Tone of Enthusiasim and make the Facts Engaging
to read. Also put effort into sounding Human rather than AI)")
            Main[f"Messier {i}"] = response
            print(f"Messier {i} has been added to the Dictionary!")
        time.sleep(1)
        print("All the Messier Object's Information has been added to
the Dictionaries!")
        time.sleep(0.4)
        print(Main)
        time.sleep(4)

    elif (inp == 3):
        print("Thanks a lot for using my Program!")
        time.sleep(1)
        print("Made with Love by Taha :D")
        time.sleep(1)
        print("Okiee Biee!")
```

```
        time.sleep(0.4)
        break
exit()
```

Database_Population_Code-Sierra.py

```
#This Program is attempting to read and add the data from a text file
that has some of the Data for the Database taken from a Wikipedia page
(https://en.wikipedia.org/wiki/List_of_discoverers_of_Messier_objects)
[PS: The Wikipedia page's Latest edit was by me!! :D]

import sqlite3 as sql
import time

Messier_Discoveries = {}
i = 0

DB = sql.connect("The_Messier_Dualis_Database.db")
Cursor = DB.cursor()

txtfile = open("MessierObjects_Discoveries.txt", "r", encoding="utf-8")
for line in txtfile:
    parts = line.strip().split(" ")
    Messier_Num = parts[1]
    Messier_Number = f"M{Messier_Num}"
    Discoverer = " ".join(parts[2:-1])
    Disco_Year = int(parts[-1])
    Messier_Discoveries[Messier_Number] = (Discoverer, Disco_Year)
print(Messier_Discoveries)
time.sleep(5)

for i in range(1, 111):
    Discoverer, Disco_Year = Messier_Discoveries[f"M{i}"]
    Cursor.execute("UPDATE The_Messier_Objects_Catalogue SET
Discovery_Date = ?, Discoverer = ? WHERE Messier_Number = ?;",
(Disco_Year, Discoverer, f"M{i}"))
    print(f"Info for Messier {i} has been added!")
    time.sleep(0.4)

print("All the Info has been added to the Database!! :D")
time.sleep(1)
print("PS: This Data was taken from a Wikipedia page whose latest Edit
if by me! :D")
```

```
time.sleep(1)
print("Anyways, Biee!")

DB.commit()
DB.close()
txtfile.close()
exit()
```

Database_Population_Code-Tango.py

```
----------------------------------------

import csv
import sqlite3 as sql
import time

print("Starting...")
Main = {}
i = 0

DB = sql.connect("The_Messier_Dualis_Database.db")
Cursor = DB.cursor()

csvfile = open("Database_Messier_Descriptions.csv", "r")
reader = csv.reader(csvfile)
headers = next(reader)
values = next(reader)
Main = dict(zip(headers, values))
time.sleep(1)

for data in Main:
i = i + 1
Value = Main[data]

    addcommand = "UPDATE The_Messier_Objects_Catalogue SET Description
= ? WHERE Messier_Number = ?;"
    Cursor.execute(addcommand, (Value, f"M{i}"))
    print(f"Messier {i} Description has been Added!")
    time.sleep(0.2)

time.sleep(1)
DB.commit()
DB.close()
csvfile.close()
print("All the Database Values have been Updated! We have a go for
```

```
Liftoff!!")
time.sleep(1)
print("Exiting...")
time.sleep(0.5)
exit()
```

Database_Population_Code-Xray.py

```
------- ----- ----- ------ ----- ----- ----- --

import json
#import Daataabase_Konnectoor as DBCON
import sqlite3 as sql

'''
Reorganized and edited version of previously found Messier Catalogue
API.
QAEZZ (https://github.com/QAEZZ/messier-api-stable) & russiandev
(https: //rep lit.c om/@ru ssian dev/m essie r - api)
MIT License Copyright (c) 2023 Osric Dienda Permission is hereby
granted, free of charge, to any person obtaining a copy of this
software and associated documentation files (the "Software"), to deal

in the Softwarewithout  restriction,including  withoutlimitationthe
rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:
The abovecopyrightnoticeandthispermission     noticeshallbe   included
in all copies or substantial portions of the Software.THE SOFTWARE IS
PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN DBECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.
BSD 3-Clause License Copyright (c) 2022, QÆZZ All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are

met: 1. Redistributions  ofsource  code mustretaintheabovecopyright
notice, this list   ofconditionsandthe   followingdisclaimer.2.
Redistributions in binary form must reproduce the above copyright
notice, this list ofconditionsandthe    followingdisclaimerin   the
documentation and/or other materials provided with the distribution. 3.
Neither the name ofthe  copyright holder northenamesof    its
contributors may be used to endorse or promote products derived from
```

```
this softwarewithoutspecificpriorwrittenpermission.      THISSOFTWARE
IS PROVIDED BYTHECOPYRIGHT   HOLDERSAND  CONTRIBUTORS\"ASIS\"   AND ANY
EXPRESS ORIMPLIEDWARRANTIES,INCLUDING,BUTNOTLIMITED       TO,THE
IMPLIED WARRANTIESOFMERCHANTABILITY   ANDFITNESSFORAPARTICULAR
PURPOSE ARE DISCLAIMED. INNOEVENTSHALLTHECOPYRIGHTHOLDER       OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
'''


#Cursor = DBCON.DBCon()
DB = sql.connect("The_Messier_Dualis_Database.db")
Cursor = DB.cursor()

jsonfile = open('Messier Database (By Osric Dienda).json', 'r')
data = json.loads(jsonfile.read())
for obj in data['data'].values():
    Cursor.execute(f'''INSERT INTO The_Messier_Objects_Catalogue
(Messier_Number, Name, Distance, Object_Type, Constellation,
Apparent_Magnitude, Image_File) VALUES (
                "M{obj['messierNumber']}", "{obj['name']}",
{obj['distance']}, "{obj['type']}", "{obj['constellation']}",
{obj['magnitude']}, "Messier_{obj['messierNumber']}.jpg"
    );''')

print("All Done!! :D")

DB.commit()
DB.close()
```

Image_Downloading_Code-Charlie.py

```
----------------------------------------

import requests as reqs
import shutil as shush
import time

URLs = "https://osricdienda.com/messier-api/images/M{num}.jpg"
for i in range(1, 111):
    print(f"Downloading Image for Messier {i}... /")
```

```
    time.sleep(0.1)
    print(f"Downloading Image for Messier {i}... |")
    time.sleep(0.1)
    print(f"Downloading Image for Messier {i}... \\")
    time.sleep(0.1)
    print(f"Downloading Image for Messier {i}... --")
    time.sleep(0.1)
    print(f"Downloading Image for Messier {i}... /")
    time.sleep(0.1)
    print(f"Downloading Image for Messier {i}... |")
    time.sleep(0.1)
    print(f"Downloading Image for Messier {i}... \\")
    time.sleep(0.1)
    print(f"Downloading Image for Messier {i}... --")
    Response = reqs.get(URLs.format(num = i), stream=True)
    f = open(f'Assets/Image Files/Messier_{i}.jpg', 'wb')
    shush.copyfileobj(Response.raw, f)
    f.close()


print("All Images successfully Downloaded! :D")
time.sleep(1)
exit()
#Thanks! :D
```

Main-Program-India.py

```
----------------------------------------

#Hieeeeee!!! 👋

import sqlite3 as sql
import pygame as pg
import keyboard as keys
import time
import random
from PIL import Image
import tkinter
import threading as MultiProcessing
import datetime

print("Hallo World!! 👋👋")
time.sleep(2)
print("Taha Husain this side!")
time.sleep(2)
print("This is my Very First Major CS Project and I am soooooo Proud of
```

```
it!")
time.sl eep(2 .5)
print("I wanted to combine My Love and Passion for Astronomy with my
Upcoming Hobby of Programming.")
time.sl eep(3 )
print("And honestly, what better way to prove it than making something
like thisformy   CSfinalproject?!")
time.sl eep(3 )
print()
print("Well anyways, Without any further Ado...")
time.sl eep(2 )
print("Welcome to Messier Dualis!")
time.sl eep(2 )
print("Enjoy! :D")
time.sl eep(1 )

CurTime = time.time()
current_datetime = datetime.datetime.now()
timestamp_string = current_datetime.strftime("%Y-%m-%d %H:%M:%S")
print(f"Current Date and Time: {timestamp_string}")
time.sl eep(1 .4)

help_inp = str(input("Do you want the Instructions on how to operate
this Software? (Yes = y; No = n): "))


if (help_inp.lower() == 'y'):
    tim e.sle ep(0. 5) pri nt() pri nt() print("Sweet! (PS: Now some
    of my work will not go unnoticed :D)



\n")
    tim e.sle ep(1. 4)
    print("There are 2 modes in the Software \n")
    tim e.sle ep(1)
    print("You can either go for Randomised Exploration mode where the
program chooses random objects for you to explore from the catalogue
\n")
    tim e.sle ep(3)
    print("And Serial-Wise Exploration mode which shows objects in the
order that Sir Charles Messier Added them in \n")
    tim e.sle ep(3)
    print("So, In the Menu Page, There is a button 'Dive In'. That
takes you to the main window \n")
```

```
    time.sleep(2)
    print("The Image of the Object will appear on the Right and info
about the Object will appear on the Left \n")
    time.sleep(2)
    print("You can move to the next object using the 'Right Arrow' key
\n")
    time.sleep(1.5)
    print("You can move to the previous object using the 'Left Arrow'
Key \n")
    time.sleep(1.5)
    print("In Addition, You can exit the program anytime by pressing
the 'Escape' Key")
    time.sleep(0.4) print("(Not Sure why you would ever wanna do that tho
    👋) \n") time.sleep(1.6) print("Their is Music and sound effects too
    for some Ear-Candy \n") time.sleep(3) print("Anyways, That's about
    it. Enjoy! :D") print() print() time.sleep(1.5)




elif (help_inp.lower() == 'n'):
    print("No worries! Welcome Back :D")
    print()
    print()
    time.sleep(2)

else:
    print("Invalid Input, Showing Instructions as default...")
    print()
    print()
    time.sleep(1.4)
    print("There are 2 modes in the Software \n")
    time.sleep(1)
    print("You can either go for Randomised Exploration mode where the
program chooses random objects for you to explore from the catalogue
\n")
    time.sleep(3)
    print("And Serial-Wise Exploration mode which shows objects in the
order that Sir Charles Messier Added them in \n")
    time.sleep(3)
    print("So, In the Menu Page, There is a button 'Dive In'. That
```

```
takes you to the main window \n")
    time.sleep(2)
    print("The Image of the Object will appear on the Right and info
about the Object will appear on the Left \n")
    time.sleep(2)
    print("You can move to the next object using the 'Right Arrow' key
\n")
    time.sleep(1.5)
    print("You can move to the previous object using the 'Left Arrow'
Key \n")
    time.sleep(1.5)
    print("In Addition, You can exit the program anytime by pressing
the 'Escape' Key")
    time.sleep(0.4) print("(Not Sure why you would ever wanna do that tho
    👋) \n") time.sleep(1.6) print("Their is Music and sound effects too
    for some Ear-Candy \n") time.sleep(3) print("Anyways, That's about
    it. Enjoy! :D") print() print() time.sleep(1.5)




breathe_scale = 1.0
breathe_direction = 1
breathe_speed = 0.00006
RandomMessierList = []
SerialExplo = 1
randomised = 0
RandomMessierNum2 = 0
Inp = int(input("Do you Want Randomised or Serial-Wise Exploration of
the Messier Catalogue? (1 = Random, 2 = Serial-Wise): "))

"""if (time.time() - CurTime >= 14):
    print("No Input Recieved. Commencing in Randomised Exploration
Mode...")
    randomised = 1"""

if (Inp == 1):
    randomised = 1
    print("Alrighty! Commencing in Randomised Exploration Mode!...")
    time.sleep(1)
    """for Taha in range (1, 111):
```

```
    RandomMessierNum = random.randint(1, 110)
    RandomMessierList.append(RandomMessierNum)"""
    RandomMessierList = random.sample(range(1, 111), 110)

elif (Inp == 2):
    randomised = 0
    print("Gotcha! Commencing in Serial-Wise Exploration Mode!...")
    time.sleep(1)
    SerialExplo = 1
elif (Inp > 2 or Inp < 1):
    print("You gave an Invalid Input, Commencing program in Serial-Wise
Exploration Mode...")
time.sleep(1)
randomised = 0

root = tkinter.Tk()
actual_screen_width = root.winfo_screenwidth()
actual_screen_height = root.winfo_screenheight()
root.destroy()

pg.init()
pg.font.init()
pg.mixer.init()

DB = sql.connect("The_Messier_Dualis_Database.db")
Cursor = DB.cursor()

STATE_INTRO = "Intro"
STATE_MAIN = "Main"
CurState = STATE_INTRO
FADEOUT_TIME = 2000
VOLUME = 2

SCREEN_WIDTH = 800
SCREEN_HEIGHT = 680
screen = pg.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT),
pg.RESIZABLE)
pg.display.set_caption("The Messier Dualis")

WHITE = (255, 255, 255)
BLACK = (0, 0, 0)
BLUE = (100, 100, 255)
CYAN = (0, 140, 220)
```

```
GREEN = (100, 255, 100)
RED = (255, 100, 100)
PURPLE = (191, 64, 191)
ORANGE = (255, 170, 0)

Background_Org = Image.open("Assets/Image Files/Background Image.jpg")

#Welcome_Text = "Welcome to The Messier Dualis!"
#Welcome_Text2 = "Ready to dive in?"
button_rect = pg.Rect(0, 0, 200, 50)
button_text = "Dive in!"
font = pg.font.Font("Assets/Nasalization.ttf", 40)
Welcome_Text = font.render("Welcome to The Messier Dualis!", True,
PURPLE)
Welcome_Text2 = font.render("Ready to dive in?", True, GREEN)

Playlist_Choice = random.randint(1, 1000)
if (Playlist_Choice >= 800):
    playlist = [
        "Assets/Audio Files/Music/Billions and Billions -
Stellardrone.mp3",
        "Assets/Audio Files/Music/Interstellar Soundtrack - Dust - Hans
Zimmer.mp3",
        "Assets/Audio Files/Music/Astronomical Objects - Paolo
Vivaladi.mp3",
        "Assets/Audio Files/Music/Meteor Showers - Paolo Vivaladi.mp3"
    ]
elif (Playlist_Choice < 800 and Playlist_Choice > 400):
    playlist = [
        "Assets/Audio Files/Music/Interstellar Soundtrack - Dust - Hans
Zimmer.mp3",
        "Assets/Audio Files/Music/Astronomical Objects - Paolo
Vivaladi.mp3",
        "Assets/Audio Files/Music/Billions and Billions -
Stellardrone.mp3",
        "Assets/Audio Files/Music/Meteor Showers - Paolo Vivaladi.mp3"
    ]
elif (Playlist_Choice <= 400):
    playlist = [
        "Assets/Audio Files/Music/Astronomical Objects - Paolo
Vivaladi.mp3",
        "Assets/Audio Files/Music/Interstellar Soundtrack - Dust - Hans
Zimmer.mp3",
```

```
        "Assets/Audio Files/Music/Billions and Billions -
Stellardrone.mp3",
        "Assets/Audio Files/Music/Meteor Showers - Paolo Vivaladi.mp3"
    ]

current_track = 0

MUSIC_END = pg.USEREVENT + 1
pg.mixer.music.set_endevent(MUSIC_END)
Click_Soundeffect = pg.mixer.Sound("Assets/Audio Files/Sound
Effects/computer-mouse-click.mp3")
Text_Soundeffect = pg.mixer.Sound("Assets/Audio Files/Sound
Effects/text-appear.mp3")
Startup_Sequence_Time = time.time()

def play_track(index):
    pg.mixer.music.load(playlist[index])
    pg.mixer.music.set_volume(VOLUME)
    pg.mixer.music.play(fade_ms=1500)

play_track(current_track)
running = True
#Show_Button_Text_Text = False
#ClickTime = 0
'''
texts = [
    ("Welcome to The Messier Dualis!", PURPLE),
    ("Ready to dive in?", GREEN),
]
Startup_Delay = [2, 4, 6]
'''
while (running == True):
    pg.event.set_grab(True)
    breathe_scale += breathe_direction * breathe_speed
    if (breathe_scale > 1.02 or breathe_scale < 0.98):
        breathe_direction *= -1

    if (CurState == STATE_INTRO):
        Background_Org = Background_Org.resize((SCREEN_WIDTH,
SCREEN_HEIGHT), Image.LANCZOS)
        Mode = Background_Org.mode
        Size = Background_Org.size
        Data = Background_Org.tobytes()
```

```
        Background = pg.image.fromstring(Data, Size, Mode)
        zoom_width = int(SCREEN_WIDTH * breathe_scale)
        zoom_height = int(SCREEN_HEIGHT * breathe_scale)

        zoomed_image = pg.transform.smoothscale(Background,
(zoom_width, zoom_height))

        screen.blit(zoomed_image, ((SCREEN_WIDTH -zoom_width)//2,
(SCREEN_HEIGHT - zoom_height)//2))
        #screen.blit(Background, (0, 0))
        current_time = time.time() - Startup_Sequence_Time
        '''
        for i, (text, color) in enumerate(texts):
            if (current_time >= Startup_Delay[i]):
                RenderText = font.render(text, True, color)
                if(i==0):
                    Render_Button_Rect =
RenderText.get_rect(center=(SCREEN_WIDTH // 2, 200))
                    screen.blit(RenderText,Render_Button_Rect)
                elif (i == 1):
                    Render_Button_Rect =
RenderText.get_rect(center=(SCREEN_WIDTH // 2, 360))
                    screen.blit(RenderText,Render_Button_Rect)
        '''

        if (time.time() - Startup_Sequence_Time >= 2):
            #Text _Sound effec t.pla y()
            RenderText =font.render("WelcometoThe    MessierDualis!",
True, PURPLE)
            #RenderText = RenderText.convert_alpha()
            #Rend erText .set_ alpha (0)
            Render_Button_Rect =
RenderText.get_rect(center=(SCREEN_WIDTH // 2, 200))
            #i = 100
            '''
            while (i < 250):
                t ime.sl eep(0 .01)
                i = i + 1
                R enderT ext.s et_al pha(i )
                screen.blit(RenderText, Render_Button_Rect)
            '''
            screen.blit(RenderText, Render_Button_Rect)
```

```
         if(time.time() -Startup_Sequence_Time >= 4):
            RenderText2 =font.render("Ready to dive in?", True, GREEN)
            Render_Button_Rect2 =
RenderText2.get_rect(center=(SCREEN_WIDTH // 2, 360))
            screen.blit(RenderText2, Render_Button_Rect2)

        for event inpg.event.get():
            if(event.type==pg.QUIT):
                running = False

            if(event.type==pg.MOUSEBUTTONDOWN):
                ifbutton_rect.collidepoint(event.pos):
                    #Show_Button_Text_Text = True
                    #Click Time()
                    Click_Soundeffect.play()
                    tim e.sle ep(1)
                    CurTime2 =time.time()
                    Next=True
                    CurState =STATE_MAIN

            if(event.type==pg.VIDEORESIZE):
                screen =pg.display.set_mode((event.w, event.h),
pg.RESI ZABLE )
                SCREEN_WIDTH,SCREEN_HEIGHT = event.w, event.h
                Background_Org=  Background_Org.resize((SCREEN_WIDTH,
SCREEN_HEIGHT), Image.LANCZOS)
                Mode=  Background_Org.mode
                Size=  Background_Org.size
                Data=  Background_Org.tobytes()
                Background=  pg.image.fromstring(Data, Size, Mode)

            if(event.type==MUSIC_END):
                pg.mixer.music.fadeout(FADEOUT_TIME)
                current_track=(current_track + 1) % len(playlist)
                play_track(current_track)

            """if (event.type==  pg.KEYDOWN):
                ifevent.key  ==pg.K_ESCAPE:
                    running =False"""

        if(keys.is_pressed('esc')):
                running = False
```

```
        #Welcome_Button_Rect =
Welcome_Text.get_rect(center=(SCREEN_WIDTH//2,SCREEN_HEIGHT // 2 -
150))
        #Welcome_Text2_rect =
Welcome_Text2.get_rect(center=(SCREEN_WIDTH//  2,SCREEN_HEIGHT // 2 ))

        if (time.time() - Startup_Sequence_Time >= 6):
            button_rect.center=  (SCREEN_WIDTH //2,  SCREEN_HEIGHT //2
+ 100)

            if(button_rect.collidepoint(pg.mouse.get_pos())):
                pg.draw.rect(screen, CYAN, button_rect,
border_ radiu s=12)
            else:
                pg.draw.rect(screen, BLUE, button_rect,
border_ radiu s=12)

            Button_Text =font.render(button_text,True,   BLACK)
            Button_Rect =
Button_ Text. get_r ect(ce nter= butto n_rec t. center)

            #screen.blit(Welcome_Text, Welcome_Button_Rect)
            #screen.blit(Welcome_Text2, Welcome_Text2_rect)
            screen.blit(Button_Text, Button_Rect)

        pg.d ispla y.flip ()

    if (CurState == STATE_MAIN):
        zoom_width = int(SCREEN_WIDTH * breathe_scale)
        zoom_height = int(SCREEN_HEIGHT * breathe_scale)

        zoomed_image=  pg.transform.smoothscale(Background,
(zoom_width, zoom_height))
        #screen.blit(Background, (0,0))
        screen.blit(zoomed_image, ((SCREEN_WIDTH -zoom_width)//2,
(SCREEN_HEIGHT - zoom_height)//2))
        '''
        z = pg.time.get_ticks() / 2000
        breathe_scale = 1.0 + 0.004 * math.sin(z)
        '''
        if (randomised == 1):
            if (Next == True):
                if (RandomMessierNum2 >= 110):
```

```
                    RandomMessierList = random.sample(range(1, 111),
110)

                Choice =RandomMessierList[RandomMessierNum2]

                View_Command =f"SELECT  Image_File FROM
The_Messier_Objects_CatalogueWHEREMessier_Number = 'M{Choice}';"
                Fact_Command  =f"SELECT  Name,Distance, Discovery_Date,
Discoverer,  Object_Type,Constellation, Apparent_Magnitude, Description
FROMThe_Messier_Objects_CatalogueWHERE   Messier_Number = 'M{Choice}';"
                #Fact_Command2=  f"SELECTDistance FROM
The_Messier_Objects_CatalogueWHEREMessier_Number = 'M{Choice}';"
                #Fact_Command3=  f"SELECTDiscovery_Date FROM
The_Messier_Objects_CatalogueWHEREMessier_Number = 'M{Choice}';"
                #Fact_Command4=  f"SELECTDiscoverer FROM
The_Messier_Objects_CatalogueWHEREMessier_Number = 'M{Choice}';"
                #Fact_Command5=  f"SELECTObject_Type FROM
The_Messier_Objects_CatalogueWHEREMessier_Number = 'M{Choice}';"
                #Fact_Command6=  f"SELECTConstellation FROM
The_Messier_Objects_CatalogueWHEREMessier_Number = 'M{Choice}';"
                #Fact_Command7=  f"SELECTApparent_Magnitude FROM
The_Messier_Objects_CatalogueWHEREMessier_Number = 'M{Choice}';"

                C ursor. execu te(Vi ew_Co mm and)
                Img = Cursor.fetchone()
                C ursor. execu te(Fa ct_Co mm and)
                Facts = Cursor.fetchone()
                # Cursor .exec ute(F act_C om mand2)
                #Fact2 = Cursor.fetchone()
                # Cursor .exec ute(F act_C om mand3)
                #Fact3 = Cursor.fetchone()
                # Cursor .exec ute(F act_C om mand4)
                #Fact4 = Cursor.fetchone()
                # Cursor .exec ute(F act_C om mand5)
                #Fact5 = Cursor.fetchone()
                # Cursor .exec ute(F act_C om mand6)
                #Fact6 = Cursor.fetchone()
                # Cursor .exec ute(F act_C om mand7)
                #Fact7 = Cursor.fetchone()
                ImageDir = Img[0]
                Messier =Image.open(f"Assets/Image Files/{ImageDir}")
                Messier =Messier.resize((SCREEN_WIDTH, SCREEN_HEIGHT),
Image.LANCZOS)
```

```
                Mode2 = Messier.mode Size2 = Messier.size Data2 =
                Messier.tobytes()             Background             =
                pg.image.fromstring(Data2, Size2, Mode2)


                def FactsWindowTK():
                    FactWindow = tkinter.Tk()
                    FactWindow.title(f"Some Facts about Messier
{Choice}")
                    FactWindow.configure(bg='black')

FactWin dow.g eomet ry(f"{ actua l_scr een_w id th//2}x {actu al_sc reen_h eight }+0
+0")
                    Name = Facts[0]
                    Dist = Facts[1]
                    DiscoYear = Facts[2]
                    DiscoPer = Facts[3]
                    ObjType = Facts[4]
                    Const = Facts[5]
                    AP_Mag = Facts[6]
                    Desc = Facts[7]
                    FactsText = tkinter.Label(FactWindow, text=f'''
This is Messier {Choice}
It's More Commonly Known Name is {Name}.
It is {Dist} Light Years away from Earth.
It was Discovered in the Year {DiscoYear}AD by {DiscoPer}.
It is a {ObjType}.
From Earth, It's visible in the Constellation of {Const}.
From Earth, It's apparent Magnitude is {AP_Mag}

Here's a Short Brief about Messier {Choice}!:
{Desc}
''', font=("Assets/Nasalization.ttf", 18), wraplength =
actual_screen_width//2 - 20, justify = 'center', fg="white",
bg="black")
                    FactsText.pack()
                    FactWindow.mainloop()

                MultiProcessing.Thread(target=FactsWindowTK,
daemon=True).start()
                Next = False
        elif (randomised == 0):
            if (Next == True):
```

```
                if (SerialExplo > 0 and SerialExplo < 111):
                    View_Command = f"SELECT Image_File FROM
The_Messier_Objects_CatalogueWHEREMessier_Number=    'M{SerialExplo}';"
                    Cur sor.e xecut e(Vie w_ Command )
                    Img = Cursor.fetchone()
                    ImageDir = Img[0]
                    Messier = Image.open(f"Assets/Image
Files/{ Image Dir}" )
                    Messier = Messier.resize((SCREEN_WIDTH,
SCREEN_HEIGHT), Image.LANCZOS)
                    Mode2    =    Messier.mode
                    Size2    =    Messier.size
                    Data2 = Messier.tobytes()
                    Background=  pg.image.fromstring(Data2,Size2,
Mode2)

                    Fact_Command2 = f"SELECT Name, Distance,
Discovery_Date, Discoverer, Object_Type, Constellation,
Apparent_Magnitude,DescriptionFROMThe_Messier_Objects_Catalogue
WHERE Messier_Number = 'M{SerialExplo}';"
                    Cur sor.e xecut e(Fac t_ Command 2)
                    Facts = Cursor.fetchone()
                    def FactsWindowTK():
                        FactWindow = tkinter.Tk()
                        Fact Windo w.con fi gure(bg ='bla ck')
                        FactWindow.title(f"Some FactsaboutMessier
{Serial Explo }")

FactWin dow.g eomet ry(f"{ actua l_scr een_w id th//2}x {actu al_sc reen_h eight }+0
+0")
                        Name = Facts[0]
                        Dist = Facts[1]
                        DiscoYear = Facts[2]
                        DiscoPer = Facts[3]
                        ObjType = Facts[4]
                        Const = Facts[5]
                        AP_Mag = Facts[6]
                        Desc = Facts[7]
                        FactsText =tkinter.Label(FactWindow,text=f'''
This is Messier {SerialExplo}
It's More Commonly Known Name is {Name}.
It is {Dist} Light Years away from Earth.
It was Discovered in the Year {DiscoYear}AD by {DiscoPer}.
```

```
It is a {ObjType}.
From Earth, It's visible in the Constellation of {Const}.
From Earth, It's apparent Magnitude is {AP_Mag}

Here's a Short Brief about Messier {SerialExplo}!:
{Desc}
''', font=("Assets/Nasalization.ttf", 18), wraplength =
actual_screen_width//2 - 20, justify = 'center', fg="white",
bg="black")
                        FactsText.pack()
                        FactWindow.mainloop()

                    MultiProcessing.Thread(target=FactsWindowTK,
daemon=True).start()
                    Next = False
                else:
                    print("The Catalogue has bee Exhausted.
Reseting...")
                    time.sleep(0.5)
                    SerialExplo = 1

        if (Next == False):
            Button_Text_Bravo = "Next ->"
            #screen.fill(BLACK)

            for event in pg.event.get():
                if (event.type == pg.QUIT):
                    running = False

                if (event.type == pg.VIDEORESIZE):
                    screen = pg.display.set_mode((event.w, event.h),
pg.RESIZABLE)
                    SCREEN_WIDTH, SCREEN_HEIGHT = event.w, event.h
                    Cursor.execute(View_Command)
                    Img = Cursor.fetchone()
                    ImageDir = Img[0]
                    Messier = Image.open(f"Assets/Image
Files/{ImageDir}")
                    Messier = Messier.resize((SCREEN_WIDTH,
SCREEN_HEIGHT), Image.LANCZOS)
                    Mode2 = Messier.mode
                    Size2 = Messier.size
                    Data2 = Messier.tobytes()
```

```
                    Background = pg.image.fromstring(Data2, Size2,
Mode2)

                if (event.type == MUSIC_END):
                    pg. mixer .musi c.fad eo ut(FADE OUT_T IME)
                    current_track =(current_track+   1)%  len(playlist)
                    pla y_tra ck(cu rrent _t rack)

                """if (event.type == pg.KEYDOWN):
                    if (event.key == pg.K_ESCAPE):
                        running = False"""

                 """if (event.type == pg.KEYDOWN):
                     if (event.key == pg.K_RIGHT):
                        Next = True
                        SerialExplo = SerialExplo + 1
                        RandomMessierNum2 = RandomMessierNum2 + 1"""

                """if (event.type == pg.KEYDOWN):
                    if (event.key == pg.K_LEFT):
                        Next = True
                        SerialExplo = SerialExplo - 1
                        RandomMessierNum2 = RandomMessierNum2 - 1"""

                if event.type == pg.ACTIVEEVENT:
                    if event.gain != 1:
                        pg.e vent. set_g ra b(True)

            if (keys.is_pressed('right') or keys.is_pressed('d')):
                Next = True
                SerialExplo = SerialExplo + 1
                RandomMessierNum2 = RandomMessierNum2 + 1
                t ime.sl eep(0 .2)

            if (keys.is_pressed('left') or keys.is_pressed('a')):
                Next = True
                SerialExplo = SerialExplo - 1
                RandomMessierNum2 = RandomMessierNum2 - 1
                t ime.sl eep(0 .2)

            if (keys.is_pressed('esc')):
                running = False
```

```
            '''
            if(time.time() - CurTime2 >= 4):
                button_rect.center = (SCREEN_WIDTH - 140, SCREEN_HEIGHT
- 60)

                if(button_rect.collidepoint(pg.mouse.get_pos())):
                    pg.draw.rect(screen, ORANGE, button_rect,
border_radius = 12)
                e lse:
                    pg.draw.rect(screen, RED, button_rect,
border_radius = 12)

                Button_Text2 = font.render(Button_Text_Bravo, True,
WHITE)
                Button_Rect2 =
Button_Text2.get_rect(center=button_rect.center)
                Button_Text2 = Button_Text2.convert_alpha()
                Button_Text2.set_alpha(160)
                screen.blit(Button_Text2, Button_Rect2)
            '''

            pg.display.flip()

DB.clos e()
pg.quit ()

print("From thebottom  of my heart, I truly hope you enjoyed my project
anddiscovereda   newspark of curiosity or love for our cosmos through
it!")
time.sl eep(4 )
print()
print("Anyways,By-Bieeeee!! 👋")
time.sl eep(1 )
exit()
#Bieeee! :D 👋
```

Main_Database_Program-Alpha.py

```
----------------------------------------

import sqlite3 as sql
import time
import os

dir = str(input("Please enter the Directory location of the Project
```

```
Folder: "))
os.chdir(dir)
db = sql.connect("The_Messier_Dualis_Database.db")
tabname = str(input("Please enter the name of the table that you wish
to work upon: "))

def CreateTable(tabname):
    cursor = db.cursor()
    create_table = f'''
    CREATE TABLE {tabname} (
        Messier_Number TEXT PRIMARY KEY,
        Name TEXT,
        Distance INTEGER,
        Discovery_Date INTEGER,
        Discoverer TEXT,
        Object_Type TEXT,
        Constellation TEXT,
        Apparent_Magnitude REAL,
        Description TEXT,
        Image_File TEXT
    );
    '''

    cursor.execute(create_table)
    db.commit()

def ShowDB(tabname):
    time.sleep(0.5)
    cursor = db.cursor()
    showDB = f"SELECT * FROM {tabname}"

    cursor.execute(showDB)

    rows = cursor.fetchall()
    column_names = [description[0] for description in
cursor.description]
    print(column_names)
    time.sleep(0.4)
    for row in rows:
        print(row)
    db.commit()

def ShowDBParticular(tabname):
```

```
    mesnum = int(input("Please enter the Messier Number that you wish
to see the Records of: "))
    time.sleep(0.5)
    cursor = db.cursor()
    showDBParticular = f"SELECT * FROM {tabname} WHERE Messier_Number=
'M{mesnum}';"

    cursor.execute(showDBParticular)

    rows = cursor.fetchall()
    column_names = [description[0] for description in
cursor.description]
    print(column_names)
    time.sleep(0.4)
    for row in rows:
        print(row)
    db.commit()

def UpdateTable(tablename, messier, name, dist, date, discoverer,
types, const, mag, desc, fun, img):
    cursor = db.cursor()

    update_table = f'''INSERT INTO {tablename}
    (Messier_Number, Name, Distance, Discovery_Date, Discoverer,
Object_Type, Constellation, Apparent_Magnitude, Description, Fun_Fact,
Image_File)
    VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
    '''

    cursor.execute(update_table, (messier, name, dist, date,
discoverer, types, const, mag, desc, fun, img))
    db.commit()

def DelRow(tablename, messier):
    cursor = db.cursor()

    del_row = f"DELETE FROM {tablename} WHERE Messier_Number =
'{messier}';"

    cursor.execute(del_row)
    db.commit()

start_time = time.time()
```

```
print("Program Starting...")
time.sleep(1)

while(True):
    print("1. Create a Table")
    time.sleep(0.5)
    print("2. Update a Record in the Table")
    time.sleep(0.5)
    print("3. Delete a Record in the Table")
    time.sleep(0.5)
    print("4. Show the Entire Table")
    time.sleep(0.5)
    print("5. Show a Particular record from the Table")
    time.sleep(0.5)
    print("6. Exit the Program")
    time.sleep(0.5)
    inp = int(input("Please enter the number of the operation that you
want to perform: "))
    time.sleep(0.5)
    print("Executing!...")
    time.sleep(0.3)

    if (inp == 1):
        CreateTable(tabname)
        time.sleep(1)
        print("Table Created! :D")

    elif (inp == 2):
        time.sleep(1)
        mesname = str(input("Please enter the Messier Number: "))
        time.sleep(0.6)
        objname = str(input("Enter the Name of the Messier Object: "))
        time.sleep(0.6)
        dist = int(input("Please enter the Distance of the Messier
Object from Earth in Light-Years: "))
        time.sleep(0.6)
        objdate = str(input("Enter the Year that the object was
discovered in (Format: YYYY): "))
        time.sleep(0.6)
        objdisc = str(input("Enter the Full Name of the Discoverer of
the Messier Object: "))
time.sleep(0.6)
objtype = str(input("Please enter the Type of the Messier
```

```
Object: "))
        time.sleep(0.6)
        constname = str(input("Please enter the name of the
Constellation in which the Messier Object is located in: "))
        time.sleep(0.6)
        apmag = float(input("Please enter the Apparent Magnitude of the
Messier Object: "))
        time.sleep(0.6)
        desc = str(input("Please enter a short Description of the
Messier Object: "))
        time.sleep(0.6)
        funfac = str(input("Please enter a short Fun Fact about the
Messier Object: "))
        time.sleep(0.6)
        img = str(input("Please enter the Filename of the Image of the
Messier Object: "))
        time.sleep(0.6)
        img2 = dir + '/' + "Image Files/" + img
        UpdateTable(tabname, mesname, objname, dist, objdate, objdisc,
objtype, constname, apmag, desc, funfac, img2)
        time.sleep(1)
        print("Table Updated! :D")

    elif (inp == 3):
        time.sleep(0.6)
        mesname = str(input("Please enter the Messier Number of the
record: "))
        time.sleep(0.6)
        DelRow(tabname, mesname)
        time.sleep(1)
        print("Editing Complete! :D")

    elif (inp == 4):
        time.sleep(1)
        ShowDB(tabname)
        time.sleep(1)
        print("Done!")

    elif (inp == 5):
        time.sleep(0.4)
        ShowDBParticular(tabname)
        time.sleep(1)
        print("Done!")
```

```
    elif(inp == 6):
        print("Exiting!...")
        time.sleep(1)
        print("Thanks a lot for using my Program!")
        time.sleep(1)
        print("Made with Love by Taha :D")
        time.sleep(1)
        break

db.clos e()
exit()
```

Python to DOCX Convertor.py

```
---------------------------------------

from docx import Document
import os

FOLDER_PATH = "." # folder with .py files
OUTPUT_FILE = "CS_Project_Code.docx"

doc = Document()
doc.add_heading("The Messier Dualis – Source Code", level=1)

for file in os.listdir(FOLDER_PATH):
    if file.endswith(".py"):
        doc.add_heading(file, level=2)
        doc.add_paragraph("-" * 40)

        with open(file, "r", encoding="utf-8") as f:
            code = f.read()
            p = doc.add_paragraph(code)
            p.style.font.name = "Consolas"

doc.save(OUTPUT_FILE)
print("Word file created successfully!")
exit()
```

Output of the Code

```
T:\Python Scripts\GitHub Repos\The-Messier-Dualis\Main-Program-India.exe
pygame 2.6.1 (SDL 2.28.4, Python 3.12.6)
Hello from the pygame community. https://www.pygame.org/contribute.html
Hallo World!! 🇮🇳 🇮🇳
Taha Husain this side!
This is my Very First Major CS Project and I am soooooo Proud of it!
I wanted to combine My Love and Passion for Astronomy with my Upcoming Hobby of Programming.
And honestly, what better way to prove it than making something like this for my CS final project?!

Well anyways, Without any further Ado...
Welcome to Messier Dualis!
Enjoy! :D
Current Date and Time: 2026-01-01 18:48:11
Do you want the Instructions on how to operate this Software? (Yes = y; No = n): n
No worries! Welcome Back :D


Do you Want Randomised or Serial-Wise Exploration of the Messier Catalogue? (1 = Random, 2 = Serial-Wise): 1
```

Some Facts about Messier 57

This is Messier 57
It's More Commonly Known Name is Ring Nebula.
It is 4100 Light Years away from Earth.
It was Discovered in the Year 1779AD by Charles Messier.
It is a Planetary Nebula.
From Earth, It's visible in the Constellation of Lyra.
From Earth, It's apparent Magnitude is 8.8

Here's a Short Brief about Messier 57!:
Messier 57, also known as the Ring Nebula, is a stunning celestial gem located in the constellation Lyra, just 2,600 light-years away from Earth. This vibrant, donut-shaped nebula showcases the remnants of a dying star, glowing with beautiful hues of blue and green. As you gaze into its depths, you witness the intricate dance of gases expelled during the star's final stages, creating a mesmerizing spectacle! The Ring Nebula is a favorite among stargazers and astrophotographers, making it a must-see for anyone eager to explore the wonders of the universe!

The Messier Dualis

Some Facts about Messier 76

This is Messier 76
It's More Commonly Known Name is Little Dumbbell Nebula.
It is 3400 Light Years away from Earth.
It was Discovered in the Year 1780AD by Pierre Méchain.
It is a Planetary Nebula.
From Earth, It's visible in the Constellation of Perseus.
From Earth, It's apparent Magnitude is 10.1

Here's a Short Brief about Messier 76!:
Messier 76, affectionately nicknamed the "Little Dumbbell," is a stunning planetary nebula located in the constellation Perseus. Spanning about 2.5 light-years across, this celestial gem captivates with its distinctive hourglass shape and vibrant hues. Formed from dying stars shedding their outer layers, it showcases fascinating structures and intricate details best seen through telescopes. At roughly 2,500 light-years away, Messier 76 reminds us of the universe's beauty and complexity, sparking wonder and curiosity among stargazers and astronomers alike!

The Messier Dualis

SQL Table Structure

```
Messier_Number TEXT PRIMARY KEY,
Name TEXT,
Distance INTEGER,
Discovery_Date INTEGER,
Discoverer TEXT,
Object_Type TEXT,
Constellation TEXT,
Apparent_Magnitude REAL,
Description TEXT,
Image_File TEXT
```

Some of the Datapoints


```
"The_Messier_Objects_Catalogue": [
  {
    "Messier_Number": "M1",
    "Name": "Crab Nebula",
    "Distance": 6300,
    "Discovery_Date": 1731,
    "Discoverer": "John Bevis",
    "Object_Type": "Supernova Remnant",
    "Constellation": "Taurus",
    "Apparent_Magnitude": 8.4,
    "Description": "Messier 1, also known as the Crab Nebula, is a breathtaking supernova remnant located in the constellation Taurus. Spannir
    "Image_File": "Messier_1.jpg"
  },
  {
    "Messier_Number": "M2",
    "Name": "NGC 7089",
    "Distance": 37500,
    "Discovery_Date": 1746,
    "Discoverer": "Jean-Dominique Maraldi",
    "Object_Type": "Globular Cluster",
    "Constellation": "Aquarius",
    "Apparent_Magnitude": 6.5,
    "Description": "Messier 2, a dazzling globular cluster located about 37,500 light-years away in the constellation Aquarius, is a stunning
    "Image_File": "Messier_2.jpg"
  },
```

Directory of the Project Folder

```
T:.
├───Assets
│   ├───Archived Shtuff
│   ├───Audio Files
│   │   ├───Music
│   │   └───Sound Effects
│   └───Image Files
├───build
│   └───Main-Program-India
│       └───localpycs
├───dist
└───__pycache__
```

```
T:\Python Scripts\GitHub Repos\The-Messier-Dualis>dir
 Volume in drive T is Local Disk
 Volume Serial Number is B816-A6D0

 Directory of T:\Python Scripts\GitHub Repos\The-Messier-Dualis

01-01-2026  17:33    <DIR>          .
01-01-2026  17:33    <DIR>          ..
01-01-2026  16:25    <DIR>          Assets
01-01-2026  16:25    <DIR>          build
01-01-2026  16:16            45,566 CS_Project_Code.docx
02-12-2025  12:35            60,326 Database_Messier_Descriptions.csv
02-12-2025  12:35             2,874 Database_Population_Code-Romeo.py
02-12-2025  12:35             1,374 Database_Population_Code-Sierra.py
02-12-2025  12:35               846 Database_Population_Code-Tango.py
02-12-2025  12:35             3,484 Database_Population_Code-Xray.py
01-01-2026  17:33               658 DB to JSON Convertor.py
01-01-2026  16:12    <DIR>          dist
01-01-2026  13:59                34 ext_reqs.txt
02-12-2025  12:35               995 Image_Downloading_Code-Charlie.py
02-12-2025  12:35             1,089 LICENSE
01-01-2026  16:11        27,157,804 Main-Program-India.exe
01-01-2026  16:00            23,621 Main-Program-India.py
01-01-2026  16:09               722 Main-Program-India.spec
02-12-2025  12:35             5,413 Main_Database_Program-Alpha.py
02-12-2025  12:35            58,034 Messier Database (By Osric Dienda).json
02-12-2025  12:35             3,874 MessierObjects_Discoveries.txt
01-01-2026  17:33            97,436 Messier_Dualis_Datapoints.json
21-12-2025  19:48               596 Python to DOCX Convertor.py
01-01-2026  17:31             3,093 README.md
02-12-2025  12:35            90,112 The_Messier_Dualis_Database.db
01-01-2026  16:25    <DIR>          __pycache__
              20 File(s)     27,557,951 bytes
               6 Dir(s)  66,121,334,784 bytes free
```

Bibliography & Credits

- Images by: NASA & Hubble Space Telescope with Jet Propulsion Laboratory
- Datapoints By: Myself; ‘call_gpt’; Wikipedia (https://en.wikipedia.org/wiki/List_of_discoverers_of_Messier_objects); Osric Dienda (https://osricdienda.com/messier-api/)

- Music:
  - Astronomical Objects - Paolo Vivaladi
  - Billions and Billions - Stellardrone
  - Dust - Hans Zimmer (Interstellar)
  - Meteor Showers - Paolo Vivaladi

- Fonts used: Nasalization (Google Fonts)
- Extra Software Used: GitHub, Visual Studio Code, Notepad, cmd, Audacity, Python (3.12.6)
- Code By: Myself :D (With Debugging help from ChatGPT)

- Python Modules Used: pygame, tkinter, pyinstaller, keyboard, call_gpt, math, random, pillow, requests, threading, time, os, sqlite3, datetime, docx, json, csv, shutil

Conclusion

Making this project told me a lot about me, It also taught me a lot of things that would otherwise be hard to realise.
I am glad i got the chance to make this project and i hope that my project helps bring the cosmos to more people. That’s it from me!

Thanks a lot for your time!